Amendments to the Claims

Claim 16 (currently amended): A computer ~~implemented~~ method for programmatically creating a distributed object program in which at least one complex object is passed as a parameter, wherein the programmatically-created program is programmatically generated from a programmer-written program which is not specially adapted for distributed execution, the ~~distributing one or more objects of a program across more than one physical device, each object containing one or more programmed member functions, said member functions having complex objects, said complex objects including one or more programmed member functions, as parameters, said~~ method comprising the computer executable steps of:

identifying ~~all of the~~ one or more objects in the programmer-written program, wherein each of the objects contains one or more programmed member functions and wherein at least one of the programmed member functions is written to pass one of the objects as a parameter;

determining a first set ~~which~~ of the identified objects which are to reside on a first computer and a second set ~~which~~ of the identified objects which are to reside on a second computer, wherein the first set and the second set together comprise the identified objects of the programmer-written program and the first set and the second set each include at least one of the identified objects; ~~such that the distributed system will consist of at least a first object on a first computer and a second object on a second computer;~~

~~——— identifying all programmed methods contained in each object that may be accessed from a remote computer;~~

programmatically generating, upon detecting that a first object in the first set contains logic to call one of the programmed member functions of a second object in the second set, a first proxy and a second proxy for ~~each~~ the second object, wherein the first proxy is generated to be installed on the first computer and the second proxy is generated to be installed on the second computer;

programmatically generating logic in the first proxy that will programmatically generate a third proxy, responsive to detecting that the call to the programmed member function of the second object will pass, as a parameter, a third object that is a complex object and that is one of the objects in the first set, wherein the third proxy is generated to be installed on the first computer; and

programmatically generating logic in the second proxy that will programmatically generate a fourth proxy, responsive to a call from the first proxy that includes a reference to the third proxy, wherein the fourth proxy is generated to be installed on the second computer,

such that, at run time, the first object can transparently access the programmed member function of the second object and the programmed member function of the second object can transparently access a programmed member function of the third object. ~~that may be accessed from a remote computer, said first proxy residing on said first computer and said second proxy residing on said second computer, said first proxy containing network linkage and indication to access programmed member functions on said second proxy on said second computer including logic to transfer and translate complex objects which reside on said first computer used as member function parameters and said second proxy containing linkage and indication to access said programmed member functions on said second object including logic to transfer and translate complex objects, said complex objects containing one or more programmed member functions and reside on said first computer, used as member function parameters; and,~~

~~accessing said remote programmed methods through said proxies.~~

Claim 17 (currently amended): ~~A method~~ The method as claimed in Claim 16, wherein:

~~said~~ the ~~logic in said~~ first proxy further comprises programmatically-generated logic to ~~on said first computer to transfer and translate complex data objects comprising the steps of:~~

~~creating a third proxy, for said complex object, which is to reside on said first computer with said complex object, said third proxy containing linkage and indication to access programmed member functions on said complex object;~~

~~creating~~ (1) create a reference table entry which correlates ~~said~~ the third proxy object to ~~said complex~~ the third object, which may be accessed by ~~said~~ the third proxy ~~object to access said complex~~ when invoking programmed member functions of the third object; (2) translate calls for the programmed member function of the second object that are received from the first object and that pass the third object as a parameter, whereby a reference to the third proxy replaces the third object on the received calls, and forward the translated calls to the second proxy; and (3) upon receiving, from the second proxy, responses to the translated calls, return the responses to the first object; ~~; and,~~

~~passing as a member function parameter to said second proxy on said second machine a reference to said third proxy, in place of said complex object when said complex object is to be a parameter in a member function call to said second object on said second machine.~~

~~said logic in said~~ the second proxy further comprises programmatically-generated logic to ~~on said second computer to transfer and translate complex data objects comprising the steps of:~~

~~creating a fourth proxy for said complex object on said first computer which is to reside on said second computer, said fourth proxy containing network linkage and indication necessary to access programmed member functions on said third proxy on said first machine;~~

~~creating a~~ (1) create a second reference table entry which correlates ~~said~~ the fourth proxy ~~to a~~ to the reference to ~~said~~ the third proxy ~~on said third computer~~, which may be accessed by ~~said~~ the fourth proxy ~~to access said~~ when forwarding calls to the third proxy; (2) invoke the programmed member function of the second object, responsive to receiving one of the translated calls that is forwarded from the first proxy, wherein the second reference table entry is consulted so that an indication of the fourth proxy is substituted, on the invocation, for the reference to the third proxy; and (3) upon receiving, from the programmed member function of the second object, a response to the forwarded call, return the response to the first proxy;

the fourth proxy further comprises programmatically-generated logic to (1) consult the second reference table entry, responsive to receiving a call from the programmed member function of the second object, thereby determining that the received call corresponds to the third proxy; (2) translate the call received from the programmed member function of the second object such that the translated call refers to the third proxy and forward the translated call to the third proxy; and (3) upon receiving, from the third proxy, a response to the translated call, return the response to the programmed member function of the second object; and

the third proxy further comprises programmatically-generated logic to (1) consult the reference table entry, responsive to receiving the translated call from the fourth proxy, thereby determining that the received translated call corresponds to the third object; (2) translate the call received from the fourth proxy to invoke the programmed member function of the third object and forward the translated call to the third object, where the programmed member function of the third object will then be executed; and (3) upon receiving, from the programmed member function of the third object, a result of the execution, return the result to the fourth proxy. ;

~~passing as a member function parameter to said second object from said second proxy on said second computer an indication of said fourth proxy, in place of said reference to said third proxy on said first computer, which represents said complex object on said first computer.~~

~~said network linkage and indication in said fourth proxy necessary to access programmed member functions on said third proxy on said first computer comprising the steps of:~~

~~looking up said fourth proxy in said reference table on said second computer to determine which object on said first machine said fourth object is a proxy for, said lookup returning a reference to said third proxy on said first computer;~~

~~calling the appropriate programmed member functions in said third proxy on said first computer.~~

~~said linkage and indication in said third proxy necessary to access programmed methods on said complex object comprising the steps of:~~

~~looking up said third proxy in said reference table on said first computer to determine which object on said first machine said third object is a proxy for, said lookup returning a reference to said complex object on said first computer;~~

~~calling the appropriate programmed member functions in said complex object.~~

Claim 18 (currently amended): ~~A method~~ The method as claimed in Claim ~~17 wherein~~ 16, wherein ~~one of said complex objects~~ the third object is ~~said~~ identical to the first object, such that the call to the programmed member function of the second object executes as a callback on the first object. ~~on said first computer.~~

Claim 19 (currently amended): ~~A method~~ The method as claimed in Claim 16, wherein ~~17~~ ~~wherein said~~ the reference table entry and the second reference table entry are created as ~~is a~~ database entries.

Claim 20 (currently amended): A computer program product for programmatically creating a distributed object program in which at least one complex object is passed as a parameter, wherein the programmatically-created program is programmatically generated from a programmer-written program which is not specially adapted for distributed execution, the ~~distributing one or more objects of a program across more than one physical device, each object containing one or more programmed member functions, said member functions having complex objects, said complex objects including one or more programmed member functions, as parameters, said~~ computer program product comprising:

a computer-readable storage medium have computer-readable program code means embodied in said medium, said computer-readable program code means comprising:

computer-readable program code means for identifying ~~all of the~~ one or more objects in the programmer-written program, wherein each of the objects contains one or more programmed member functions and wherein at least one of the programmed member functions is written to pass one of the objects as a parameter;

computer-readable program code means for determining a first set ~~which~~ of the identified objects which are to reside on a first computer and ~~which~~ a second set of the identified objects which are to reside on a second computer, wherein the first set and the second set together comprise the identified objects of the programmer-written program and the first set and the second set each include at least one of the identified objects; ~~such that the distributed system will consist of at least a first object on a first computer and a second object on a second computer;~~

~~computer-readable program code means for identifying all programmed methods contained in each object that may be accessed from a remote computer;~~

computer-readable program code means for programmatically generating, upon detecting that a first object in the first set contains logic to call one of the programmed member functions of a second object in the second set, a first proxy and a second proxy for ~~each~~ the second object, wherein the first proxy is generated to be installed on the first computer and the second proxy is generated to be installed on the second computer;

computer-readable program code means for programmatically generating logic in the first proxy that will programmatically generate a third proxy, responsive to detecting that the call to the programmed member function of the second object will pass, as a parameter, a third object that is a complex object and that is one of the objects in the first set, wherein the third proxy is generated to be installed on the first computer; and

computer-readable program code means for programmatically generating logic in the second proxy that will programmatically generate a fourth proxy, responsive to a call from the first proxy that includes a reference to the third proxy, wherein the fourth proxy is generated to be installed on the second computer.

such that, at run time, the first object can transparently access the programmed member function of the second object and the programmed member function of the second object can transparently access a programmed member function of the third object. ~~that may be accessed from a remote computer, said first proxy residing on said first computer and said second proxy residing on said second computer, said first proxy containing network linkage and indication to access programmed member functions on said second proxy on said second computer including logic to transfer and translate complex objects which reside on said first computer used as member function parameters and said second proxy containing linkage and indication to access said programmed member functions on said second object including logic to transfer and translate complex objects, said complex objects containing one or more programmed member functions and reside on said first computer, used as member function parameters; and,~~

~~computer-readable program code means for accessing said remote programmed methods through said proxies.~~

Claim 21 (currently amended): ~~A computer~~ The computer program product as claimed in Claim 20, wherein:

~~said logic in said~~ the first proxy ~~on said first computer to transfer and translate complex data objects comprising the steps of:~~

~~creating a third proxy, for said complex object, which is to reside on said first computer with said complex object, said third proxy containing linkage and indication to access programmed member functions on said complex object;~~

~~creating~~ further comprises programmatically-generated logic to (1) create a reference table entry which correlates ~~said~~ the third proxy ~~object~~ to ~~said complex~~ the third object, which may be accessed by ~~said~~ the third proxy ~~object to access said complex~~ when invoking programmed member functions of the third object; (2) translate calls for the programmed member function of the second object that are received from the first object and that pass the third object as a parameter, whereby a reference to the third proxy replaces the third object on the received calls, and forward the translated calls to the second proxy; and (3) upon receiving, from the second proxy, responses to the translated calls, return the responses to the first object; ~~; and;~~

~~passing as a member function parameter to said second proxy on said second machine a reference to said third proxy, in place of said complex object when said complex object is to be a parameter in a member function call to said second object on said second machine.~~

~~said logic in said~~ the second proxy ~~on said second computer to transfer and translate complex data objects comprising the steps of:~~

~~creating a fourth proxy for said complex object on said first computer which is to reside on said second computer, said fourth proxy containing network linkage and indication necessary to access programmed member functions on said third proxy on said first machine;~~

~~creating a~~ further comprises programmatically-generated logic to (1) create a second reference table entry which correlates ~~said~~ the fourth proxy ~~to a~~ to the reference to ~~said~~ the third proxy ~~on said third computer~~, which may be accessed by ~~said~~ the fourth proxy ~~to access said~~ when forwarding calls to the third proxy; (2) invoke the programmed member function of the second object, responsive to receiving one of the translated calls that is forwarded from the first proxy, wherein the second reference table entry is consulted so that an indication of the fourth proxy is substituted, on the invocation, for the reference to the third proxy; and (3) upon receiving, from the programmed member function of the second object, a response to the forwarded call, return the response to the first proxy;

the fourth proxy further comprises programmatically-generated logic to (1) consult the second refcrence table entry, responsive to receiving a call from the programmed member function of the second object, thereby determining that the received call corresponds to the third proxy; (2) translate the call received from the programmed member function of the second object such that the translated call refers to the third proxy and forward the translated call to the third proxy; and (3) upon receiving, from the third proxy, a response to the translated call, return the response to the programmed member function of the second object; and

the third proxy further comprises programmatically-generated logic to (1) consult the reference table entry, responsive to receiving the translated call from the fourth proxy, thereby determining that the received translated call corresponds to the third object; (2) translate the call received from the fourth proxy to invoke the programmed member function of the third object and forward the translated call to the third object, where the programmed member function of the third object will then be executed; and (3) upon receiving, from the programmed member function of the third object, a result of the execution, return the result to the fourth proxy. ~~;~~

~~passing as a member function parameter to said second object from said second proxy on said second computer an indication of said fourth proxy, in place of said reference to said third proxy on said first computer, which represents said complex object on said first computer.~~

~~said network linkage and indication in said fourth proxy necessary to access programmed~~

~~member functions on said third proxy on said first computer comprising the steps of:~~

~~looking up said fourth proxy in said reference table on said second computer to determine which object on said first machine said fourth object is a proxy for, said lookup returning a reference to said third proxy on said first computer;~~

~~calling the appropriate programmed member functions in said third proxy on said first computer.~~

~~said linkage and indication in said third proxy necessary to access programmed methods on said complex object comprising the steps of:~~

~~looking up said third proxy in said reference table on said first computer to determine which object on said first machine said third object is a proxy for, said lookup returning a reference to said complex object on said first computer;~~

~~calling the appropriate programmed member functions in said complex object.~~

Claim 22 (currently amended): ~~A computer~~ The computer program product as claimed in Claim ~~21 wherein one of said complex objects~~ 20, wherein the third object is ~~said~~ identical to the first object, such that the call to the programmed member function of the second object executes as a callback on the first object. ~~on said first computer.~~

Claim 23 (currently amended): ~~A computer~~ The computer program product as claimed in Claim ~~21 wherein said~~ 20, wherein the reference table ~~is a~~ entry and the second reference table entry are created as database entries.

Claim 24 (currently amended): A computer system for programmatically creating a distributed object program in which at least one complex object is passed as a parameter, wherein the programmatically-created program is programmatically generated from a programmer-written program which is not specially adapted for distributed execution, the ~~distributing one or more objects of a program across more than one physical device, each object containing one or more programmed member functions, said member functions having complex objects, said complex objects including one or more programmed member functions, as parameters, said~~ system comprising:

means for identifying ~~all of the~~ one or more objects in the programmer-written program, wherein each of the objects contains one or more programmed member functions and wherein at least one of the programmed member functions is written to pass one of the objects as a parameter;

means for determining ~~which~~ a first set of the identified objects which are to reside on a first computer and a second set ~~which~~ of the identified objects which are to reside on a second computer, wherein the first set and the second set together comprise the identified objects of the programmer-written program and the first set and the second set each include at least one of the identified objects; ~~such that the distributed system will consist of at least a first object on a first computer and a second object on a second computer;~~

~~means for identifying all programmed methods contained in each object that may be accessed from a remote computer;~~

means for programmatically generating, upon detecting that a first object in the first set contains logic to call one of the programmed member functions of a second object in the second set, a first proxy and a second proxy for ~~each~~ the second object, wherein the first proxy is generated to be installed on the first computer and the second proxy is generated to be installed on the second computer;

means for programmatically generating logic in the first proxy that will programmatically generate a third proxy, responsive to detecting that the call to the programmed member function of the second object will pass, as a parameter, a third object that is a complex object and that is one of the objects in the first set, wherein the third proxy is generated to be installed on the first computer; and

means for programmatically generating logic in the second proxy that will programmatically generate a fourth proxy, responsive to a call from the first proxy that includes a reference to the third proxy, wherein the fourth proxy is generated to be installed on the second computer,

such that, at run time, the first object can transparently access the programmed member function of the second object and the programmed member function of the second object can transparently access a programmed member function of the third object. ~~that may be accessed from a remote computer, said first proxy residing on said first computer and said second proxy residing on said second computer, said first proxy containing network linkage and indication to access programmed member functions on said second proxy on said second computer including logic to transfer and translate complex objects which reside on said first computer used as member function parameters and said second proxy containing linkage and indication to access said programmed member functions on said second object including logic to transfer and translate complex objects; said complex objects containing one or more programmed member functions~~

~~and reside on said first computer, used as member function parameters; and,~~

~~means for accessing said remote programmed methods through said proxies.~~

Claim 25 (currently amended): ~~A system~~ The system as claimed in Claim 24, wherein:

~~said logic in said~~ the first proxy ~~on said first computer to transfer and translate complex data objects comprising the steps of:~~

~~creating a third proxy, for said complex object, which is to reside on said first computer with said complex object, said third proxy containing linkage and indication to access programmed member functions on said complex object;~~

~~creating~~ further comprises programmatically-generated logic to (1) create a reference table entry which correlates ~~said~~ the third proxy ~~object~~ to ~~said complex~~ the third object, which may be accessed by ~~said~~ the third proxy ~~object to access said complex~~ when invoking programmed member functions of the third object; (2) translate calls for the programmed member function of the second object that are received from the first object and that pass the third object as a parameter, whereby a reference to the third proxy replaces the third object on the received calls, and forward the translated calls to the second proxy; and (3) upon receiving, from the second proxy, responses to the translated calls, return the responses to the first object; ~~and,~~

~~passing as a member function parameter to said second proxy on said second machine a reference to said third proxy, in place of said complex object when said complex object is to be a parameter in a member function call to said second object on said second machine.~~

~~said logic in said~~ the second proxy ~~on said second computer to transfer and translate~~

~~complex data objects comprising the steps of:~~

~~creating a fourth proxy for said complex object on said first computer which is to reside on said second computer, said fourth proxy containing network linkage and indication necessary to access programmed member functions on said third proxy on said first machine;~~

~~creating a~~ further comprises programmatically-generated logic to (1) create a second reference table entry which correlates ~~said~~ the fourth proxy ~~to a~~ to the reference to ~~said~~ the third proxy ~~on said third computer~~, which may be accessed by ~~said~~ the fourth proxy ~~to access said~~ when forwarding calls to the third proxy; (2) invoke the programmed member function of the second object, responsive to receiving one of the translated calls that is forwarded from the first proxy, wherein the second reference table entry is consulted so that an indication of the fourth proxy is substituted, on the invocation, for the reference to the third proxy; and (3) upon receiving, from the programmed member function of the second object, a response to the forwarded call, return the response to the first proxy;

the fourth proxy further comprises programmatically-generated logic to (1) consult the second reference table entry, responsive to receiving a call from the programmed member function of the second object, thereby determining that the received call corresponds to the third proxy; (2) translate the call received from the programmed member function of the second object such that the translated call refers to the third proxy and forward the translated call to the third proxy; and (3) upon receiving, from the third proxy, a response to the translated call, return the response to the programmed member function of the second object; and

the third proxy further comprises programmatically-generated logic to (1) consult the reference table entry, responsive to receiving the translated call from the fourth proxy, thereby determining that the received translated call corresponds to the third object; (2) translate the call received from the fourth proxy to invoke the programmed member function of the third object and forward the translated call to the third object, where the programmed member function of the third object will then be executed; and (3) upon receiving, from the programmed member function of the third object, a result of the execution, return the result to the fourth proxy.

~~passing as a member function parameter to said second object from said second proxy on said second computer an indication of said fourth proxy, in place of said reference to said third proxy on said first computer, which represents said complex object on said first computer:~~

~~said network linkage and indication in said fourth proxy necessary to access programmed member functions on said third proxy on said first computer comprising the steps of:~~

~~looking up said fourth proxy in said reference table on said second computer to determine which object on said first machine said fourth object is a proxy for, said lookup returning a reference to said third proxy on said first computer;~~

~~calling the appropriate programmed member functions in said third proxy on said first computer:~~

~~said linkage and indication in said third proxy necessary to access programmed methods on said complex object comprising the steps of:~~

~~looking up said third proxy in said reference table on said first computer to determine which object on said first machine said third object is a proxy for, said lookup returning a reference to said complex object on said first computer;~~

~~calling the appropriate programmed member functions in said complex object:~~

Claim 26 (currently amended): ~~A system~~ The system as claimed in Claim ~~25 wherein one of said complex objects~~ 24, wherein the third object is identical to the ~~said~~ first object, such that the call to the programmed member function of the second object executes as a callback on the first object. ~~on said first computer.~~

Claim 27 (currently amended): ~~A system~~ The system as claimed in Claim ~~25 wherein said~~ 24, wherein the reference table entry and the second reference table entry are created as ~~is a~~ database entries.